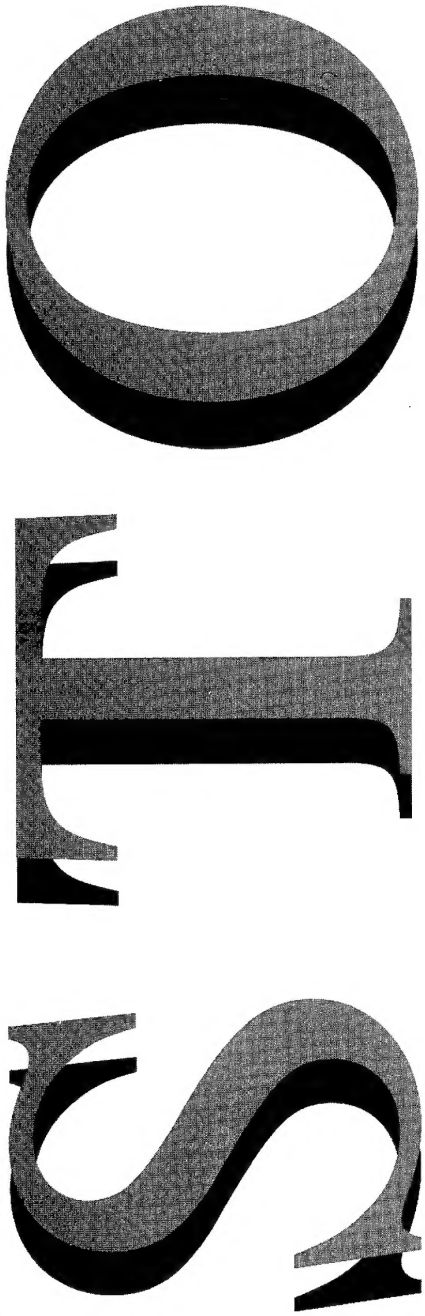
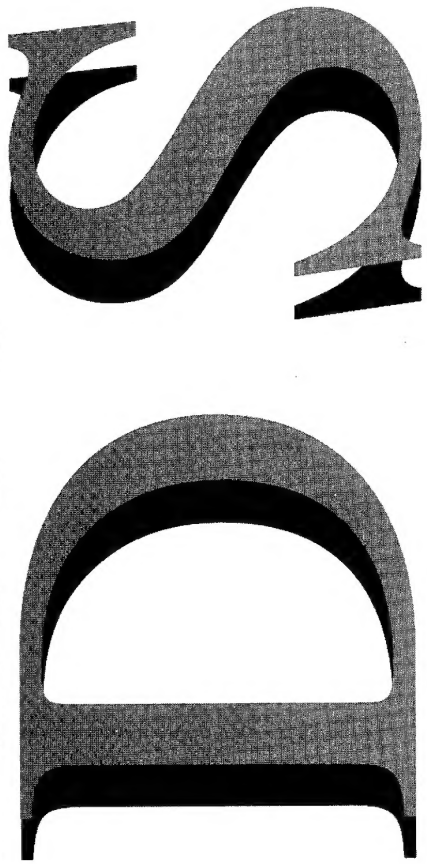
AR-010-201



Requirements Management Tools Evaluation
User Needs and Evaluation Criteria

Andrew P. Gabb, Neelan Maheswaran
and Alan M. Allwright

DSTO-GD-0139



APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Requirements Management Tools Evaluation
User Needs and Evaluation Criteria

Andrew P. Gabb, Neelan Maheswaran and Alan M. Allwright

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-GD-0139

ABSTRACT

This paper discusses the users' needs and evaluation criteria for requirements management (RM) tools for use in front end systems engineering processes in the Australian Defence Organisation. Part 1 of the paper describes the user needs for RM tools; Part 2 describes the evaluation criteria, based on those needs, to be used for comparative evaluation of the tools. The purpose of the paper was to establish the basis for a systematic evaluation of possible tools.

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108

Telephone: (08) 8259-5555
Fax: (08) 8259-6567

AR-010-201
August 1997

APPROVED FOR PUBLIC RELEASE

Requirements Management Tools Evaluation User Needs and Evaluation Criteria

Executive Summary

This paper discusses the users' needs and evaluation criteria for requirements management (RM) tools for use in front end systems engineering processes in the Australian Defence Organisation (ADO). The purpose of the paper was to establish the basis for a systematic evaluation of possible tools.

This paper has been divided into two almost independent parts because of the advantages in describing the needs and criteria in different contexts. Part 1 describes the user needs for requirements management tools, while Part 2 addresses the evaluation criteria derived from those needs. This separation of needs and criteria encouraged a more objective evaluation, and provides readers external to the ADO with criteria which are easier to understand.

Part 1 - User Needs

Part 1 discusses user needs for RM tools in the context of the potential users in the ADO, including typical activities and specifications. These needs have been obtained from a number of sources including surveys of projects in the ADO and the authors' own experiences in defence acquisition. Early drafts of this paper were also circulated to several different projects and project support agencies, and discussed with potential tool users.

The use of RM tools in the ADO is increasing, as is the understanding and appreciation of the importance of getting the requirements right and managing them in a disciplined manner. In the pre-contract phases of a project, the requirements can change quickly, and this volatility calls for more automation in how requirements are represented and managed so that the effects of requirements changes can be assessed. However, there are wide variations in the capabilities of different RM tools, and selection of the right tool is an important decision which can affect the whole project.

In examining the needs for RM tools, we firstly looked at the typical types and formats of specifications and other statements of needs, the specification development activities, the environments in which these documents are developed, and the background of the staff (the potential tool users) carrying out these activities. We then developed numerous scenarios, cases, viewpoints and individual needs which might arise in the capture, definition and management of needs.

Part 2 - Evaluation Criteria

The evaluation criteria are directly derived from the needs identified in Part 1. They address desirable tool features, functions and performance such that different RM tools can be assessed and compared against the criteria.

Broadly, the criteria cover the following:

- The information that the tool manages with respect to requirements, including attributes (additional information), links to other requirements, numbering of requirements and special formatting.
- Importing requirements into the tool, including the automatic 'parsing' of existing document based specifications, and exporting requirements to other tools.
- Managing different versions of requirements and specifications (configuration management) and checking for consistency of linking.
- The generation of specification documents in different formats.
- The reporting of different information about the set of requirements being managed, including metric data.
- Tool usability including the flexibility in displaying data.
- The flexibility and efficiency in creating, deleting and modifying requirements and their associated data.
- Performance issues including capacity and speed issues.
- Access control for different users.
- Miscellaneous issues including tool administration, the computer environment in which it runs, technical support and reliability.

Authors

Andrew P. Gabb
Information Technology Division

***Andrew Gabb** has been with the Australian Defence Science and Technology Organisation since graduating in Electrical Engineering and Computing Science at the University of Adelaide in 1971. In that time he has acted as a consultant to numerous major projects in the Australian Department of Defence, in the areas of systems and software engineering primarily for combat systems. He is currently a Principal Research Scientist conducting research into C3I front end systems engineering and procurement methods.*

Neelan Maheswaran
Information Technology division

***Neelan Maheswaran** has been with Australian Defence Science Technology Organisation (DSTO) since 1990. Soon after graduating in Electrical Engineering at the University of Adelaide in 1985, he worked for the Submarine Warfare Systems Centre (SWSC). During his time at SWSC he was a member of a team developing and maintaining software for the Oberon Class Submarine Fire Control System. Since joining DSTO, he has undertaken studies for Army on the computerisation of the Artillery Fire Control System and also developed a concept demonstrator. His current research interests include C3I front end systems engineering and requirements management.*

Alan M. Allwright
Information Technology division

***Alan Allwright** graduated in Mathematics and Computing from the South Australian Institute of Technology in 1988 and received a Masters in Computing Science from the University of South Australia in 1995. Since starting work with the Defence Science and Technology Organisation in 1989 Alan has provided assistance to a number of Defence projects on issues related to combat systems.*

Contents

Abbreviations

ADO	Australian Defence Organisation
CASE	Computer aided system/software engineering
CDRL	Contract data requirements list
DOR	Detailed operational requirements
FPS	Functional and performance specification
HTML	Hypertext markup language
INCOSE	International Council on Systems Engineering
ODBC	Open database connectivity
OLE	Object linking and embedding
OPEVAL	Operational evaluation
PC	'IBM compatible' personal computer
RM	Requirements management
RFT	Request for tender
RFP	Request for proposal
SOW	Statement of work
SQL	Structured query language
TBD	To be determined

PART 1 - USER NEEDS

1. INTRODUCTION

This paper discusses the users' needs and evaluation criteria for requirements management (RM) tools for use in front end systems engineering processes in the Australian Defence Organisation (ADO). Part 1 of the paper describes the user needs for RM tools; Part 2 describes the evaluation criteria, based on those needs, to be used for comparative evaluation of the tools. The purpose of the paper was to establish the basis for a systematic evaluation of possible tools.

At the front end of the acquisition life cycle, the most important task is to understand the users' needs, and translate them into user requirements. A skilled system engineer is required to capture, analyse and model the available information from a myriad of sources into a organised set of requirements. Often this is an iterative process that will result in many alterations to the initial set of requirements. The emergent and volatile nature of requirements necessitates control and tracking of the evolution of requirements from their origin, by means of requirements management. Manual handling of information in this iterative process can become cumbersome and tedious as the volume of information increases and changes emerge. It can also lead to errors.

The needs identified in this paper have been obtained from a number of sources including surveys of projects in the ADO and the authors' own experiences in defence acquisition. Early drafts of this paper were also circulated to several different projects and project support agencies, and discussed with potential tool users.

The users' needs addressed here apply only to the *definition and management* of requirements. It is assumed that other requirements engineering activities, such as requirements capture, analysis and validation, are essentially separate activities that will not strongly influence tool selection.

This paper has been divided into two almost independent documents because of the advantages in describing the needs and criteria in different contexts. Part 1 of this paper necessarily discusses user needs in the context of the potential users in the ADO, including typical activities and specifications. Part 2, which is derived from the needs in Part 1, addresses desirable tool features, functions and performance in a more general sense. This encouraged a more objective evaluation, and provides readers external to the ADO with criteria which are easier to understand.

2. NEED FOR REQUIREMENTS TOOLS

2.1 The importance of requirements management

The following is an excerpt from a report dated 24 June 1994 titled *Analysis of Automated Requirements Management Capabilities* developed for Rome Laboratory, US Air Force Materiel Command by Software Productivity Solutions:

> The development of large, complex systems presents many challenges to system engineers. Foremost among these are the ability to ensure that the final system satisfies the needs of the users and provides for easy maintenance and enhancement of these systems during their deployed lifetime. These systems often change and evolve throughout their life cycle, making it difficult to track the implemented system against original and evolving user requirements.
>
> Requirements establish an understanding of the user's needs, and also provide the final yardstick against which implementation success is measured. Requirements management consists of information capture, information storage and management, and information dissemination. Information management is at the heart of the requirements management problem and includes organization, traceability, analysis and visualization. Key to the success of any requirements management process is requirements traceability. Requirements traceability is a technique used to provide relationships between requirements, design and implementation of a system in order to manage the effect of change and ensure the success of the delivered systems. A recent survey of NCOSE [now INCOSE - the International Council on Systems Engineering] members identified improved support for requirements traceability as the most critical need to be addressed by automated systems engineers' tools. For requirement traceability to be effective, it is necessary to associate requirements with information stored in a variety of system engineering tools.

Requirements management tools are already being used in the front end system engineering processes in Defence. In many cases the tool is a modern general purpose database tool such as Microsoft Access. There is, however, a growing interest in and use of specialised RM tools. There is no doubt that this usage will grow in the future.

2.2 The importance of getting requirements right

It is now widely accepted that the majority of problems in the development of complex systems stem from incorrect or inadequate requirement specifications. Studies of large projects have also shown that more than half of the system errors are introduced in the requirements definition phase.

Such errors are difficult to detect. Some are revealed during the design process when it becomes clear that the emerging design, although compliant with the system specification in the contract Statement of Work, will not satisfy the users' needs. Others are detected during acceptance testing or operational evaluation

(OPEVAL). Still more are found during operation of the system in service. In addition, the errors can be very costly to fix, not only because they can affect the architectural design of the system, but also because they are found so late in the development life cycle.

2.3 Requirements management in the pre-contract phases

Requirements are most volatile in the pre-contract phases of a project. The activities of requirements capture, analysis, definition and validation are normally carried out iteratively, with new requirements discovered, or existing requirements modified and refined, frequently over a relatively long period of time. Where a large complex system is the objective of the requirements, the problems of maintaining consistency in specifications containing hundreds or even thousands of requirements becomes increasingly difficult. Recent Navy specifications for major ships and systems have had over 1000 requirements in the RFT (Request for Tender) specifications alone.

A specification document cannot provide sufficient information to manage requirements adequately, and still fulfil all the needed functions of a specification. Information could be included showing the rationale for each requirement, the traceability between different requirements and other pertinent information, but the specification would quickly become unreadable. If this information is stored separately, the risks of inconsistency between the requirements and the associated information is likely to become unacceptably high.

Storing the requirements in a database appears to be the best solution to this problem. The specification can then be generated from the database when required.

Including information in a database also allows many other functions to be provided which may improve the efficiency and effectiveness of requirements management. Some of these include:

- Being able to detect requirements which have not been traced to lower requirements.
- The inclusion of references to documents and meetings where decisions to change requirements were made.
- The ability to keep several versions of requirements.
- Maintaining an automatic audit trail of who changed a requirement and when.
- Support for and compatibility with computer based tender evaluation tools.
- Being able to see the requirements potentially affected when a higher level requirement changes.
- Collecting metrics of requirements development, to be used both for process improvement and task estimation in later projects.
- Providing reports on many aspects of the requirements and their associations.

2.4 The downside of RM tools

There are copious examples of RM tools becoming "shelfware" when it was found that the tools did not meet the immediate needs of the users, or where the users' requirements engineering process was incompatible with the tools. There is no doubt that RM tools require more discipline in their use than more traditional requirements engineering practices where requirements are simply recorded using a wordprocessor.

There have also been numerous complaints about the usability of such tools, with suggestions that the advantages the tools provide do not outweigh the difficulty and effort in using them. Whether these are fair comments is debatable. The fact that systems engineers will strongly resist using tools they are unhappy with is undeniable.

Specification development is often on the critical path in the pre-contract phases of a project. If there is any likelihood that using an RM tool will hazard the timely delivery of the specifications, the tool is doomed.

For this reason, the usability of RM tools is considered the most important aspect in this evaluation.

3. OVERVIEW OF SPECIFICATION DEVELOPMENT

3.1 Specifications

This section summarises the levels and types of specifications and other statements of requirement which need to be managed. The list is indicative only; the types of specifications, their formats and names will vary with the relevant arm of Defence (Navy, Air Force, Army, Joint) and the application. Some specifications, such as the Functional and Performance Specification, may consist of several discrete but related specification documents (e.g. for a ship platform, the combat system, and logistics support).

3.1.1 Detailed Operational Requirements (DOR)

The DOR is a detailed performance based statement of requirement, usually derived from higher level documents including an Operational Concept (opconcept) Document. The DOR will continue to be developed after contract signature, and will usually be used as the basis for acceptance into service (not to be confused with contractual acceptance). The DOR is typically developed by Development Division, who will sponsor the project, and is written in the language of the users and user representatives.

3.1.2 Functional and Performance Specification (FPS)

The FPS is a performance based specification to be used in the tendering and contract process. The FPS is normally developed under the control of the Project Director (the acquirer). While it is derived from the DOR, many of the requirements are at a similar level. The FPS is sometimes referred to as an "engineering" or "technical" specification, because it pays more attention to the development and production aspects than higher level specifications, and, although it is a user specification, it's language is likely to reflect the fact that it must be understood by developers.

3.1.3 Contractual requirements

The contractual requirements will include the Statement of Work (SOW), specifications and Contract Data Requirements List (CDRL) agreed by the acquirer, sponsor and system supplier. The specifications are often at varying levels of detail, possibly including operational requirements, functional and performance requirements, one or more system and subsystem specifications, and product and process definitions. The contractual requirements are the basis for design and development by the supplier.

3.1.4 Formats

The format for the DOR, FPS and contractual specifications is similar to that of A and B-Specs defined in MIL-STD-490A *Specification Practices*. Other documents may be descriptive in style, sometimes using sequentially numbered paragraphs to the JSP(AS)102 standard (see below for more detail).

3.2 Activities

The following are seen as critical activities. Again, the activities are indicative rather than typical.

3.2.1 Develop and validate Detailed Operational Requirements

The activity is to derive, capture, analyse, define and validate requirements for the DOR. To some extent the DOR will be derived from higher level documents, but much additional requirement capture and analysis will be necessary.

3.2.2 Develop and validate Functional and Performance Specification

The activity is to derive, capture, analyse, define and validate the requirements for the FPS. To some extent the FPS will be derived from the DOR, but much additional requirement capture and analysis will be necessary, particularly in analysing and ensuring the feasibility of requirements. The FPS is used as the requirements baseline for the Request for Proposal (RFP) or Request for Tender (RFT).

The main difference between a Request for Proposal (RFP) and Request for Tender (RFT) is in the level of the specifications and the detail in the corresponding offers from potential suppliers. Several large ADO projects in recent times have employed an RFP phase to shortlist suppliers, followed by a more detailed RFT phase.

3.2.3 Evaluate RFP or RFT responses

The tender evaluation criteria are based on the FPS requirements. As a condition of tendering, the tenderers are required to provide a statement of compliance against the FPS and other RFP/RFT requirements, and traceability to their proposals.

The technical evaluation activity will result in a number of issues for each proposal to be addressed in contract negotiation (as a by-product in addition to supplier selection).

3.2.4 Contract technical negotiation

The technical negotiations will culminate in technical specifications included in contractual requirements, which need to be traceable to the FPS (and ultimately back to the users' higher level requirements). The contractual requirements will include test requirements.

4. ENVIRONMENT AND ASSUMPTIONS

4.1 Environment and users

Relevant factors include:

a. The acquirer's staff are likely to be the heaviest users of the tool.
b. The tool user base is seen as follows: operational users, support staff, engineering and logistic staff. Few have extensive requirements engineering training or experience. All have basic literacy in window based office tools.
c. All acquirers have ready access to IBM compatible networked PCs, although some also have access to Macintosh and Unix systems. While the PC is the preferred platform for hosting the tool, multi-platform tools will also be considered.
d. The number of on-line users is likely to be 4 (typical) to 10 (maximum). Many others will contribute to the requirements engineering activities, including the off-line definition of requirements. If used for tender evaluations the maximum number of users may increase to 100.
e. Multi-user tool use across a network is assumed, although single user operation will also be common, particularly in the early stages of a project.

4.2 Assumptions

The following assumptions (constraints) are made:

a. All levels of specification described below will need to be catered for by the tool.
b. Tool use will start with one of the specification types shown below.
c. In terms of the cases specified below, the period from development of the operational concept (opconcept) to the letting of the contract is considered the most relevant. Maintenance of the DOR and higher requirements (by the project sponsor) and lower level requirements (by the acquirer and/or supplier) may continue during development, but this is not considered to be critical to the tool selection.
d. Use of the tool is likely to occur in bursts, with periods of intense activity followed by periods of inactivity (while the specifications are being reviewed, for example).

4.3 Notable omissions

The most notable omission in the above is the need for requirements modelling and analysis functions. In our opinion, such functions would rarely be used if incorporated in an RM tool. This is not to say that modelling and analysis does not occur. In many cases complex analyses will be some of the most important inputs to requirements. However, at the levels of requirements considered, the relationship between the requirements and the models is usually relatively simple and can be expressed with a few links. Similarly, it will be rare for the same staff to carry out the modelling and to manage the requirements.

We believe that the prime requirement for the tool is the management of requirements rather than analysis, and in most cases there are superior analysis and modelling tools available. We therefore see any modelling capability as a bonus rather than a basis for purchase and use.

5. SCENARIOS, CASES, VIEWPOINTS AND NEEDS

This section presents scenarios, cases (mini-scenarios) and tool user viewpoints expressing the needs for RM tools for pre-contract requirements management. These are *not* formal requirements and should not be read as such. No attempt has been made to prioritise the cases, and the reader should be wary not to interpret the language as a genuine indication of priority. For example, some of the needs expressed as "should" may eventuate as mandatory requirements.

The following cases are not exhaustive, and the inclusion of a need in a scenario does not imply that the need is exclusive to that scenario. In fact, this is rarely

the case. Nor should the reader infer that this section illustrates typical practice or views in the Australian Defence Organisation. However, we (the authors) are confident that the situations and views presented here have existed, or will exist, in customer-based requirements engineering for complex systems.

5.1 Starting from scratch

a. A specification is required for a new system. Little is known about the detailed requirements and requirements capture activities are about to start. It is likely that there will be many false trails and differing opinions leading to high volatility in the requirements.
b. The rationale for requirements will come from various sources including internal communications, trade studies, policy and planning documents, minutes of meetings, scenarios (such as these), even telephone calls. We would like to identify these sources in the database in a unique way, so that we can find out, for example, what other requirements were raised or changed in a particular meeting.
c. Similarly, some requirements may trace to the selection of a product or to changes in the operational users' processes, i.e. they may not need to be met by the supplier. We need to be able to describe and use such requirements "sinks".
d. We will be using a standard format, with some headings predefined. We need to quickly insert a template for this type of document.
e. We have been told that our specification structure, developed during the RE phase, is ideal for another project which will use the same tool. We therefore need to transfer the structure to another RM database using the same tool, and strip out unwanted information.

5.2 Starting with predefined requirements

a. The DOR and higher level documents exist, as specifications in wordprocessor form, and the FPS development activities are about to start. The DOR needs to be entered into the tool as the first step, semi-automatically if possible. Limited information is available about the rationale for some requirements, and no direct traceability to higher documents exists. These will need to be determined and entered later.
b. Many of the requirements will be copied verbatim from the DOR to FPS, at least at first (and may be modified later). With others, any assistance in generating a number of similar derived requirements from the same parent would be appreciated. Having to re-enter similar or identical information for each child is something we would seek to avoid.
c. We need to keep a track of which requirements have been flowed down to the FPS completely, recognising that some will be partly flowed down only. It would be useful in this latter case to write notes to ourselves and others about what still needs to be done to complete the flow down.
d. At any stage we need to be able to identify those requirements which have not been flowed down (and which should be) both onscreen and using

printed reports. Similarly, we need to be able to see which requirements do not have parents, and which are not identified as root requirements.

e. The activity is likely to introduce some new (root) requirements, many of which will need to be reflected in the DOR. Although ideally all requirements should stem from the DOR or higher, some of the requirements are "engineering" requirements, and it is more honest to include them as root requirements in the FPS.

f. Our experience shows there are 20 or more (often informal) versions of the specification produced, as the FPS development progresses. We need to make the specification quickly and easily, with minimal user interaction and post processing of the document.

g. There are many stakeholders (perhaps 40 or 50), of various backgrounds and persuasions, who need to review the FPS, or parts thereof. Most of these won't have the RM tool installed. Some will insist on a printed specification. Others will prefer an "soft" copy in Word 6.0. We may decide to "publish" the document on our internal web site, so HTML might be a useful representation. Certainly the odd Unix user would prefer this to Word 6.0.

h. Many stakeholders will only need to browse the database, and will not use the tool if it is not intuitive to use. They will not need exposure to many of its complexities, and for these tool users the tool should present a "simple face".

i. Our sponsors, who wrote the DOR, will review and eventually endorse the FPS. (They are the real owners of the requirement.) It would be useful to be able to collect their comments (and those of others) in the tool, and to show which requirements are currently not endorsed. We need to be able to mark requirements which have changed since a particular version, both in the tool and in the specification. If we don't, they may read the whole specification again, and there are obvious risks if this occurs.

j. In some cases suggestions for changes to requirements and other comments will be provided to us in a soft form, such as email, wordprocessor or spreadsheet. We need to be able to cut and paste from these documents into the RM tool.

5.3 Starting with requirements from other tools

a. The DOR and higher level documents exist, as specifications in a database, spreadsheet or a different RM tool, and the FPS development activities are about to start. Where possible, we need to enter the DOR and higher documents into the tool as the first step, with conversion of some data where necessary.

b. We do not want to lose any auxiliary information which may be included in the data. We will need to import the requirements and their attributes, and to be able to expand on this richness in our own tool.

5.4 Mixing and matching requirements

a. We are developing 3 related specifications in parallel. We had carefully planned which requirements were included in which specification, but we were wrong, again. We now need to transfer requirements, including whole branches, between the specifications. We can do this easily in a wordprocessor, including the changes in numbering, and it should be even easier in our RM tool.

b. We have found that there are some common requirements which need to be included in several specifications in the same project. Although we try to avoid duplication, in this case it is unavoidable without changing the whole specification tree. We would prefer that the requirements are defined in one place only, so that changes will affect all the specifications.

c. Specification developers are geographically separate, but working on the same specification using the same RM tool. They meet regularly (every few days) to form a consolidated version of their work to that point. While they are generally working in different areas of the specification, there is potential for overlap. Facilities need to be provided to assist in merging the two versions of the database. These should include the capability to compare databases and identify the differences between them.

d. Our requirements development staff often change roles. Few of them in any case are continuously working on requirements development. We need to be able to quickly transfer the tool from one workstation to another, including the configuration in use.

e. The system will be used at many separate sites with different but overlapping requirements. Users at each site have independently identified the requirements from their own viewpoint, which now need to be consolidated into a overall statement of requirements. The requirements will usually be in a wordprocessor format.

5.5 Clause numbering of requirements

a. We need to use a relatively strict format for our specifications, similar to, but not identical to a MIL-STD-490A A-spec, with hierarchical numeric clause numbering, e.g. 3.4.2.1, 4.1(a), 6.5.2.1(a)(1). Different elements of our organisation may use similar but different numbering systems.

b. It is possible that the DOR and other documents may use other formats including our Defence Writing standard JSP(AS)102. This standard has chapters, paragraph numbers (typically prefixed by the chapter number, e.g. "513" for the 13th paragraph in Chapter 5. It also has 3 levels of unnumbered headings.

c. We will also need to refer to Annexes and Appendices, with numbers like A.4.2.1.

d. In the early stages of specification development, we need to have requirements automatically numbered, as they are inserted, with automatic renumbering of other requirements as appropriate.

e. Later, when other documents may refer to this specification, we need to be warned if requirements numbers may change as the result of an addition,

so we don't accidentally change numbers. This will need to be done on a specification by specification basis, i.e. some specifications will have "fixed" numbering, other changeable.

f. We need to be able to create a numbering cross-reference between different versions of the specification, showing old and new numbers.

g. Some requirements will refer to others, in the same and different specifications (e.g. "see also DOR 4.2.1"). Tracking these cross-references would be very difficult manually - the tool should do this automatically.

5.6 Unique numbering of requirements

a. In some cases we will need to use a unique requirement ID number in addition to clause numbering, and to use this number in our specifications. We will also need to generate a cross-reference between the requirement ID and the clause numbering in different specifications.

5.7 Configuration management

a. In an evolutionary or phased acquisition the requirements will change as the project progresses, as the users gain experience in the use of early releases of the system. We need to know what capabilities will be met in each release, to manage changes to the requirements, and to plan subsequent phased or incremental releases. It is necessary to maintain at least 5 versions of the requirements, because we are:
   - Under warranty provisions with at least one release, and collecting problem reports.
   - Monitoring the development of another.
   - Negotiating the capability of the next phase.
   - Planning the capability of the following 2 phases.

b. We will often identify an interim version as we release a specification for internal review. This may sometimes happen on an almost weekly basis. In many cases, the value of having this version information can become negligible a few weeks later, and in fact can result in forest of versions. We would like to be able to "remove" version information from the database, as if that version never occurred.

c. We need to print specifications for different versions, including superseded versions.

d. During development the DOR will continue to be revised so that it reflects the users' operational requirements at any time. This will often result in some differences between the requirements reflected in the contractual specifications and the DOR. We will probably handle this by maintaining two versions of the DOR at this stage.

5.8 Multiple suppliers and specifications

We have decided to separately address different aspects of our system under two or more separate contracts. The high level requirements are still integral, however. This will require separate specifications to be produced under

different version control (e.g. just because we need to upgrade the version number of one specification, doesn't mean the others should change.)

5.9 Access, review and audit

a. We have several tool users entering requirements for the same specification. For consistency and accountability reasons, we wish to limit what they can change to several branches of the tree (say). They should be able to see other information, however. Some users, with limited experience, also need to be restricted to only changing some fields or attributes. It should be also be possible to prevent them deleting requirements.

b. Often, several users will need identical access rights. It should be possible to arrange this without setting up the individual rights for each user, probably using access groups.

c. Some specifications need to be frozen to prevent accidental modification, regardless of user access rights.

d. On the other hand, sometimes we will want to modify the database in a single user mode, and in this case we don't want the hassle of having to wade through a series of access checks every time we want to do something.

e. It would be useful to collect history information of how the number of requirements has changed. This should include history on the number of requirements meeting specific user defined criteria, such as orphan and spinster requirements, or unprocessed requirements.

f. Similarly, information on which user has made the changes, and when, should be automatically registered in the database. Where formal changes have been made to a defined baseline, the previous requirement information should be recorded and retrievable.

5.10 Attributes

a. We are going to need a number of attributes attached to each requirement, these will include:
   - Whether a requirement is "shall", "should" or informational.
   - Security: classification, rationale for classification, authorisation.
   - Who is the owner of the individual requirement, and who can change it.
   - Who last changed the requirement, why, and when.

b. We are certain there are more of these, and some of them will change from project to project, maybe from specification to specification within the same project. We can't define them all now, however. We need to be able to define our own, with different data types including text, numbers, dates, look-up lists (where the set of values is defined by us), and yes/no.

5.11 Special formatting

a. Our technical specifications commonly include mathematical and Greek symbols such as the degree (°) and micro (μ) symbols. The tool must support these. Similarly it needs to provide subscripts and superscripts, and text which includes words or phrases with bold, italic or underline highlighting.

b. We need to be able to generate typical specifications. While it is not essential that the specifications look identical to those we have developed in wordprocessors, they need to contain the same information, and be no less readable. Typical features include:
   - Titles, table of contents, and other fore matter.
   - Glossary of abbreviations.
   - The ability to do conditional formatting including for example the forcing of an odd page for each section.
   - Showing the classification of each clause by e.g. "(C)" for Confidential (portion marking).

c. We need to include the following in our specifications:
   - Several text tables describing existing interface protocols. We would like to be able to edit these within the tool.
   - Graphs containing required performance envelopes.
   - Illustrations (pictures) of existing equipment, for information.
   - Captured screen layouts showing the style of user interfaces of associated existing workstations, with a requirement for compatibility.
   - Engineering drawings of the layout of the area in which the system is to be installed.
   - Mathematical equations of varying complexity.

5.12 Making changes

a. Once the specification becomes relatively mature, we need to assess the effect on the requirements database as a whole when we change a higher level requirement. Similarly, we want to see why a requirement exists. To do this we need to see all the ancestors of the requirement at a glance.

b. We will often want to change one or more attributes consistently over a group of requirements. For example, we may want to change the owner or the security classification of a number of requirements. We can do this very easily in our spreadsheet tools.

5.13 Usability

a. We often mark various pages in a specification we are working on with Post-It stickers or paper clips, and quickly flip between different sections. We also tend to use a printed copy to check things out when we are working with a different part of the specification on the screen. The ability to see several parts of the specification at once is important to us.

b. We find that the Table of Contents, particularly in the early stages, is essential to seeing the structure of the specification. We also sometimes draw schematic diagrams of the specifications to show how the different areas are arranged and related. We need something similar in the tool.

c. Our experience with customised database tools has not been good, particularly for structured data. Too often we can't see where we are in the structure, and we get the feeling that not enough information is shown on the screen, compared with using a wordprocessor. Often we want to see several records together, but can only see one at a time, including a plethora of detritus that we have no interest in, at least for what we are currently doing. Even when the designer has used continuous forms, there are too few displayed on the screen. We could show you what we want (at least today) in about 5 minutes.

d. Finding our way round the database reminds us of selecting a "open file" dialog in Windows 3.1 for some applications. We seem to be repeating the same things over and over again.

e. Using the on-line help makes us think that it was written by (and for) software engineers, or possibly computer scientists. We wouldn't need the on-line help if we were computer experts. The manuals are often not much better.

f. Some of the other staff are terrible at spelling. We must have a spelling checker. Their grammar is not real good, either, but we don't trust grammar checkers anyway.

g. We are concerned over suggestions that the RM database may have several specifications in it, and that this will usually slow us down. Most of the time, we are only working on one specification, or even a limited part of it. We don't want to know about the other specifications.

5.14 RFP and RFT

a. Many of our potential suppliers want our specifications and associated information in electronic form, for rapid incorporation into their own RM tools. We need to be able to export the data in an appropriate form. For varying reasons, we will not provide them with the full database and will need to produce a subset of the specifications and include only some of the attributes. We may need to repeat the same export several times.

b. Similarly, the selected contractor will want to import parts of our database into tools used during development.

5.15 Tender evaluations

a. We often use internally developed tender evaluation tools, which need to use the requirements in one or more specification as the basis for the criteria. Although the minimum need would be to copy the requirements from the RM database to the evaluation database, it would be useful to access the requirements directly, so that the evaluation tree could be developed in conjunction with the RM tool.

b. Our evaluation tool needs are quite similar to our RM tools in some regards (and quite different in others). They are not detailed in this document. We are interested in how well an RM tool might meet our evaluation needs, and will assess this as well.

5.16 Training and support

a. Our users will need training appropriate to their particular needs. Because most of them will use the tool only occasionally, customised computer based training, or self paced training in other forms, would be very useful. Short training courses from the tool vendor's representatives may also be required.
b. We have had some unfortunate experiences with technical support of specialist computer based tools, particularly from off-shore suppliers. It is not satisfactory to have to wait for weeks for response to a question, when we have a deadline in a few days time.

5.17 Miscellaneous

a. We have had unfortunate experiences with networked database tools, including conflicts between different users changing data, "lock-ups" where no-one could access the data, and system crashes where we lost a day's work. We are looking for reliability in this area, and the ability to reconstruct work done in case of a system failure.
b. Although some of our specifications are security classified, many are not, and often only a few clauses are classified. We would value features which allow us to separate classified from unclassified data, both logically (so that the user can only view unclassified data, for example), and physically, where the classified and unclassified sections of the data are in separate databases.
c. It is likely that in the future we will need to use the RM tool in conjunction with other CASE tools, particularly configuration management tools. We would like a tool which has features which facilitate integration with other tools.

PART 2 - EVALUATION CRITERIA

1. INTRODUCTION

This Part describes criteria to be used as the basis for the comparative evaluation of requirements management (RM) tools for use in front end systems engineering processes in the Australian Defence Organisation (ADO). It is derived from the needs expressed in Part 1, which also outlines the environment in which the tools are likely to be used and user characteristics.

It is not expected that the complete capability addressed by the criteria will be uniquely satisfied by the tool in isolation (although many criteria refer to "the tool"). It will generally be acceptable if some of the functionality is provided by other common commercially available software (such as a wordprocessor), used in conjunction with the tool. It is also accepted that some configuration of the tool may be necessary to meet some requirements.

Finally, those experienced with the use of RM tools in a development environment may have difficulty in understanding some of the priorities assigned to criteria in this paper. Although many of the requirements are similar for an acquirer, they are not identical - the tools will be used in a different environment, by differently skilled personnel with different expectations than might apply to developers. The number of requirements in the tool will typically be lower, for example, and be less formally expressed than at the lower branches of the specification hierarchy.

2. NOMENCLATURE

2.1 Requirements

While this paper mainly refers to "requirements", and the objective of using the tool is requirements management, the tool must cater for a variety of information elements with most (if not all) the same attributes as requirements. These information elements include:

- Requirements, including verification requirements.
- SOW (Statement of Work) activities.
- External documents, including trade studies and analysis reports.
- Products and product development requirements.
- Clauses in operational concept documents.

- Elements of scenarios.
- Verification descriptions.

2.2 Requirements sets

The term "requirements sets" is used to refer to groups of related requirements managed by the tool including specifications and other documents.

2.3 Slices

"Slice" is used to refer to a subset of the total requirements managed by the tool. The slice is defined according to a set of criteria, often specified by the tool user.

2.4 Context

"Context" is used to refer to a subset of the total requirements currently under consideration by the user, either in the tool overall (referred to as "general context") or in a view (referred to as the context of a view). The context will normally be defined by a slice.

2.5 Views

A "view" is a visual representation of part of the RM database on the screen. This may be text based or graphical and is typically shown in a window.

2.6 Priorities

The priorities for the criteria have the following meanings.

P1	**Critical**	Essential features. Failure against this criterion means that the tool has serious limitations.
P2	**Important**	Features which are important to the effectiveness and efficiency of the tool for most applications, and which need to be integrated within the tool. The functionality cannot be easily provided in some other way.
P3	**Relevant**	Features which add real value for most applications. It is acceptable that these functions can be achieved by reasonably efficient work arounds.
P4	**Useful**	Features which will be useful in less typical applications, or which would be used less frequently.

3. REQUIREMENTS DATA

3.1 Requirement text

a. In addition to the basic text of the requirement, does the tool support qualifying text, such as clarification material or guidance, which would be closely associated with the requirement in the specification? *[P2]*
b. Is any restriction placed on the length of the requirements text? For example, the tool will need to support "long" requirements (including qualifying text), as well as more verbose information elements as described in section 2.1. *[P2]*

3.2 ID fields and numbering

3.2.1 Hierarchical numbering

a. Is support for hierarchical numbering (e.g. 3.4.1) and lists (e.g. a., b., ... and (1), (2), ...) provided? *[P1]*
b. Is there flexibility in the numbering styles? *[P2]*
c. Can the numbering be assigned automatically, with renumbering of existing requirements where necessary? *[P2]*
d. Can automatic renumbering be protected, by means of a warning and the ability to override the protection? *[P3]*
e. Is automatic cross referencing to hierarchical numbering supported in the requirement text and attributes? This applies to "see" and "see also" references to other requirements and external references. *[P2]*

3.2.2 Unique numbering

a. Can unique numbers be automatically assigned to requirements in addition to hierarchical numbering? Numbers which have been used should not be reused. *[P2]*
b. Can these unique numbers be entered from another tool (possibly with systematic conversion to prevent conflicts)? *[P3]*

3.3 Attributes

3.3.1 Attribute support

Does the tool support the following attributes? *[P1]*
a. Rationale for requirements and other comments.
b. Rationale for changes.
c. Whether a requirement is completely flowed down or not.
d. Identification of root and "sink" requirements.
e. Level of prescription (e.g. mandatory, non-mandatory, information, header).
f. Security classification.

g. Classification of the requirement, e.g. by category, type.
h. Identification of the person, group or organisation which owns, changes the requirements.
i. Date and time of creation and change.

3.3.2 User defined attributes

a. Can we define new attributes with different types? *[P1]*
b. Do the types include text, numeric, date, logical (true/false), set group of values (enumeration literal)? *[P2]*
c. Can we define validation checks for user defined attributes? *[P2]*
d. Is the number of user defined attributes limited? It is likely that at least 40 user defined attributes will be needed. *[P2]*

3.4 Links (traceability)

a. Are links provided to show both the source or sources of a requirement (why the requirement exists), and also requirements derived from this requirement? *[P1]*
b. Is M:N bidirectional linking provided? This requirement may be derived from more than one source requirement. This requirement may result in more than one derived requirement. *[P1]*
c. Can we link to and from requirements in other requirements sets (specifications)? *[P1]*
d. Can we link to and from requirements in external documents and references? *[P2]*

3.5 Special data formats

3.5.1 Symbols and formatting

a. Can we use common mathematical and Greek symbols such as the degree (°) and micro (μ) symbols? *[P2]*
b. Can we use subscripts and superscripts? *[P2]*
c. Can we use formatted text which includes words or phrases with bold, italic and underline highlighting? *[P3]*

3.5.2 Specification generation

a. Does the tool support the inclusion of information relevant to the generation of specifications including the following?
   (1) Titles, table of contents, and other fore matter. [P2]
   (2) Glossary of abbreviations. *[P2]*

3.5.3 Tables and graphics

Are the following supported?

a. Text tables. *[P2]*
b. Mathematical equations of varying complexity. *[P2]*
c. Graphs. *[P3]*
d. Pictures. *[P3]*
e. Engineering drawings. *[P3]*

4. REQUIREMENTS INPUT, OUTPUT, GENERATION, AND COMPATIBILITY

4.1 Parse and import requirements from other documents

a. Can we easily import data from text documents including wordprocessors and ASCII (or similar) text files? The parsing required is separation into clause number, security classification (if any) and text, plus an indication if the requirement is a "shall" or "should". *[P3]*
b. Can we set defaults for the attributes of imported requirements? *[P3]*

4.2 Database compatibility

This applies particularly to popular database standards such as Access, Paradox, dBase, Oracle, Ingres. The use of open database standards and protocols such as SQL client/server and ODBC would contribute to meeting these needs.

a. Can we easily import requirements from databases and spreadsheets, including attribute information? *[P1]*
b. Can we easily export requirements to databases and spreadsheets, including attribute information? *[P1]*
c. Is import/export with other RM tools supported? *[P3]*
d. Is compatibility with other CASE tools, particularly configuration management tools, supported? *[P3]*
e. Is the tool's database directly compatible with open database standards, so that we can use the database in other tools? For example, wordprocessor merge functions, direct manipulation by a database tool. *[P3]*

4.3 Support for linking and decomposition activities

a. Does the tool provide simple and efficient means of manually linking existing requirements in the same or different specifications? *[P1]*
b. Does the tool support the activity of decomposing requirements, where the user is deriving requirements typically for a lower level specification? *[P2]*

4.4 Support for tender evaluation

a. Will the tool support tender evaluation? *[P3]*

The evaluation model and tree will be based on one or more specifications, with the evaluation criteria based on individual requirements. It is highly unlikely that any tool will meet the complete needs of the evaluation activities, because of the necessity to tailor the evaluation model to the organisational needs and the specific application and acquisition strategy. However, the ability to use the tool to develop criteria, and provide traceability from evaluation criteria and requirements, could be very useful. It is critical that requirements can be exported from the RM tool to the evaluation tool. Complete integration of RM and evaluation tools is not required.

5. MANAGEMENT OF REQUIREMENTS

5.1 Managing multiple baselines

a. Can we identify and maintain different baselines? *[P2]*
b. Can we maintain different baselines for different requirement sets? *[P2]*
c. Can we review changes on the basis of the date/time that they occurred? *[P2]*
d. Can we trace a requirement through different versions of requirements sets? *[P3]*
e. Can we manage alternative (as opposed to historical) versions of one or more requirements sets? *[P3]*
f. Can we maintain a history of RM metrics? (e.g. total requirements, changes, orphans and spinsters, against user defined criteria or attributes.) *[P4]*

5.2 Change control

a. Does the tool automatically record automatically record when changes were made, and who made them? *[P2]*
b. Can we automatically maintain a history of formal changes? *[P2]*
c. Can we distinguish between formal and informal changes? Informal changes, for example, might be changes made between formal baselines. *[P3]*

5.3 Working with separate databases

a. Does the tool support merging of requirements from different databases? *[P2]*

b. Can we easily create a new database containing a subset of the requirements and attributes? The main aim is to provide a subset (some requirements only, some attributes only) to users who do not need the full database. The new database should have all excess information completely removed, i.e. not retrievable by any means. *[P2]*
c. Can we identify the differences between two databases created using the tool? *[P3]*
d. Can we physically separate the database into classified and unclassified parts, and work on the unclassified part without the need to access the classified database? *[P3]*

5.4 Link management

a. Does the tool provide for the review of missing links? *[P1]*
b. Can the tool detect inconsistencies in links? For example, circular linking or linking in the "wrong" direction in the hierarchy. *[P1]*

6. SPECIFICATION GENERATION

a. Can we generate specifications in standard wordprocessor formats (particularly Word 6.0)? *[P1]*
b. Do we have a high level of control over the generated document style? *[P2]*
c. Can we both specify the security classification of each clause by a requirement attribute and show it in specifications, e.g. "(C)" for Confidential (portion marking)? *[P2]*
d. Can we include the special formatting including graphics and tables addressed in section 3.5? *[P2]*
e. Can we generate large and complex specifications quickly with minimal manual post-processing of the specification? *[P2]*
f. Can we specify conditional formatting including for example the forcing of an odd page for each section? *[P3]*

7. REPORTING

7.1 Types of report

Many of the functions addressed in this paper (e.g. differences between versions) are likely to produce reports, or at least be more useful if the results can be represented in a report. Rather than specify actual report types, the evaluation will compare the general reporting capability. Specific areas of consideration will include:

a. Status reports, showing counts, distributions, calculations based on the status of requirements or a slice of requirements. *[P3]*
b. Impact reports, showing e.g. the consequences of changing one or more requirements. *[P2]*
c. Anomaly reports, showing requirements without ancestors or descendants, database inconsistencies, areas which are incomplete or TBD. *[P1]*
d. Comparison reports, showing difference between versions, new or changed requirements. *[P2]*
e. Analysis reports, showing different slices of the database in different formats. *[P3]*
f. Subset reports, listing requirements and a subset of attributes for a given slice. *[P3]*

7.2 Reporting flexibility

a. Do we have a high level of control over what information is printed? *[P2]*
b. Can we request reports on selected slices? *[P2]*
c. Can we preview reports on the screen before they are printed? *[P2]*
d. Can we generate reports to a text file? *[P2]*
e. Can we generate reports to a word processor? *[P3]*
f. Do we have a high level of control over the format of printed information? *[P3]*
g. Are graphical reports provided (e.g. charts and graphs)? *[P3]*
h. Can we print an equivalent of each screen view? *[P3]*

8. USABILITY

8.1 General

a. How easy is the tool to learn and use, particularly for commonly used functions such as browsing, navigation, finding, adding requirements manually, changing requirements? *[P1]*
b. Does the tool support use by novices for simple functions? Users with no experience of the tool should find it very easy to see the requirements outline, zoom to a requirement, and search with no tuition. *[P1]*
c. Does the tool support use by moderately experienced but not expert users for modifying requirements and attributes, establishing slices and generating specifications and reports? For example, consider the intuitive nature of these and similar functions? *[P1]*
d. Does the tool support power users, by e.g. keyboard shortcuts and a high level of user interface configurability? *[P2]*

8.2 Controls

a. Can the user interface be easily modified to assign functions to buttons, hot keys and menus? *[P2]*
b. Can macros be easily defined and assigned to buttons, keys and menus? *[P2]*
c. Where user defined enumeration types are used for attributes, can the user "pick" the value required from a list? *[P3]*

8.3 Views

8.3.1 General

a. Can views of different slices be viewed simultaneously? *[P2]*
b. Can different views of the same data be viewed simultaneously (e.g. outline, detail)? *[P2]*
c. Are views updated as the underlying data changes? *[P3]*

8.3.2 Types of views

Are the following views supported?
a. An outline of the requirement set hierarchy. *[P1]*
b. An outline of the requirement hierarchy within the requirement set. *[P1]*
c. Individual requirements showing attributes? *[P1]*
d. Lists of requirements (where a number of requirements can be viewed as a scrollable list)? *[P1]*
e. Can we view the preamble to a numbered list of requirements while viewing and changing individual requirements? This particularly applies to the lead-in words such as "The requirements for ... are as follows:", and other cases where the actual requirement out of context has little or no meaning. *[P2]*
f. Graphical views of hierarchies. *[P3]*
g. Graphical view of links:
   (1) Showing ancestors and descendants. *[P3]*
   (2) Within the current slice. *[P4]*
h. N^2 (or similar) chart of links between requirements sets. *[P3]*

8.3.3 Control of views

a. Can we control the level displayed in outlines? *[P2]*
b. Can we easily configure the information shown in views, particularly for the attribute data of requirements (e.g. configuration of database forms and lists)? *[P2]*
c. Can we easily alternate between different view configurations (e.g. changing the form used to display records)? *[P2]*
d. Can we easily constrain the context of a view (e.g. a list) without changing the context of other views? *[P2]*
e. Is editing in graphical views supported? *[P4]*

8.4 Navigation

a. Can we simply define a general context under which all subsequent actions will be constrained (see section 2)? Parameters could include requirements sets, baselines, branches, or external documents. *[P2]*
b. Can we easily move to a selected detail from outline and graphical views? *[P2]*
c. Can we define and use annotated bookmarks for navigation purposes? *[P2]*
d. Can the position of a requirement (in the hierarchy) be easily seen when viewing requirements or other details? *[P2]*

8.5 Miscellaneous

a. How useful and usable is the guidance provided on the use of the tool? This includes comprehensiveness, quality and ease of use of manuals and on-line help. *[P2]*
b. Can functions and configuration be defined by tool users with minimal training? While the ability to define complex functions such as queries using scripts is useful, casual users should be provided with the capability to develop such functions using more intuitive methods such as wizards, builders and macro recording. *[P3]*
c. How well does the tool support cooperation of multiple users, including rewrite, markups, annotations, approval, release, voting? *[P3]*
d. Can the user elect that slow activities such as report generation and printing be done in the background? If an activity is likely to take more than one minute (say), it should be possible for this to be done in the background, i.e. the user can continue with other activities. *[P4]*

9. FUNCTIONS

9.1 Queries & slices (filtering and sorting)

a. Is there a high level of flexibility in the selection of slices and the order in which they are displayed in views? It should be possible to select and sort on the basis of combinations of attributes, links, hierarchical numbers, ID numbers, parts of fields, priority, baseline/s, requirements type, etc. *[P1]*
b. Can complex slice and sorting criteria be defined easily by the tool users during operation (as opposed to during configuration)? *[P2]*
c. Can user defined criteria be saved, copied and edited? *[P2]*

9.2 Finding and replacing

a. Can finding and replacing be done on the following?

(1) Any text in text fields. *[P1]*
(2) Whole fields. *[P1]*

b. Can the search be applied to:
(1) Single fields. *[P1]*
(2) All fields. *[P1]*
(3) Groups of fields. *[P3]*

c. Can a slice be defined as part of the find/replace definition? This is for reasons of efficiency, i.e. it should not be necessary to query the full database when the user is trying to find a particular requirement. *[P2]*

d. Are short cuts provided for "find again" and "find/replace again"? *[P2]*

9.3 Creation, deletion and modification of requirements

a. Can we reverse operations which change the database, including deletion of data? *[P1]*

b. Can we delete a group of requirements with single action? For example, the currently defined slice for a view. *[P2]*

c. Can we easily prune (delete) and graft (move) branches in the requirements hierarchy? *[P2]*

d. Can we easily create a new requirement by copying an existing requirement, including its attributes? *[P2]*

e. Can we modify specific attributes of a designated group of requirements with a single action? For example, setting one of more attributes of a slice of requirements to the same values. *[P2]*

9.4 Editing

a. Can we cut and paste to and from other applications? *[P1]*

b. Can we cut and paste between fields within the tool? *[P1]*

c. Can we edit text tables within the tool? This may include remote editing using e.g. OLE. *[P3]*

10. PERFORMANCE ISSUES

10.1 Capacity, response

a. The number of requirements is likely to be 1000 (single requirements set), 10,000 (typical maximum requirements) and 30,000 (absolute maximum). Can the tool perform effectively with these numbers? *[P1]*

b. Is the response time adequate for straightforward control functions in the following situations? *[P1]*
(1) With a lightly loaded network and database.
(2) With many requirements and links.
(3) With multiple users accessing the database.

10.2 Speed

How do the efficiency of the tools compare, particularly for the following activities?

a. Specification generation. *[P2]*
b. Filtering. *[P2]*
c. Sorting. *[P2]*
d. Complex reports. *[P3]*
e. Start up. *[P4]*

11. PRIVILEGES AND ACCESS

11.1 User access control

In all cases below, access control includes the separate rights to view data and to change data.

a. Can the access of users to specific requirements sets be controlled? *[P1]*
b. Can the access of users to specific requirements be controlled? *[P2]*
c. Can access rights be easily set up and edited? This refers to features which may assist in this regard, such as access groups, access on the basis of attributes etc. *[P2]*
d. Can access to specific attributes be controlled? *[P2]*
e. Can users easily select access modes to restrict their own ability to make accidental changes? For example, to select "browse" mode. *[P3]*
f. Can the tool operate with all access restrictions removed (by users with the appropriate rights)? In some situations it will be appropriate for users to use the tool without the aggravations of overbearing access control. It should be possible to run the tool in a mode where the access control does not reduce efficiency. *[P3]*

11.2 Security control

a. Are specific facilities provided for the handling of classified information? This is in addition to user access control. *[P3]*

12. TOOL ADMINISTRATION

12.1 Minimising loss of data

a. Can we "close down" the database at any time and guarantee database consistency? *[P3]*
b. Can we repair a corrupted database? *[P3]*

c. Can we archive the data while users are using the data? *[P3]*
d. Can we identify which users are currently using the tool? *[P4]*

13. MISCELLANEOUS

13.1 Platform

a. Does the tool provide the full functionality on an IBM PC or compatible computer? *[P2]*
b. Is multi-platform support provided, on Unix and Macintosh computers on the same network? *[P3]*

13.2 Computer environment

a. How well does the tool coexist with the normal computer configuration and operations? This is a measure of the disruption to other applications which might be experienced by users and includes any special configuration options required. *[P2]*
b. How do the tools compare with regard to computer and network resources required? This includes memory, disk, CPU required. *[P3]*

13.3 Support of common standards

a. How well does the tool support common specification standards? This is an examination of the suitability of the tool and provided templates for common systems and software engineering standards and their derivative formats including EIA 632, IEEE 1220, MIL-STD-490A, MIL-STD-498, IEEE software engineering standards and ISO 12207. *[P2]*
b. How easy is it to tailor the provided templates to meet other (but similar) standards. *[P2]*

13.4 Word proofing

a. Is spell checking provided (including the capability to use custom dictionaries)? *[P2]*
b. Are other proofing tools provided which may improve quality? Examples might include the use of data dictionaries and abbreviation lists which provide automatic proofing of requirements sets. *[P3]*

13.5 Technical support

a. How responsive and effective is the technical support likely to be in Australia? This will depend on capability offered by vendors, and

information from other sources, including the responsiveness during the evaluation, and established track record with Australian customers. It should be based on the response to deeper technical questions and problems, rather than the simpler problems which might be solved by reading the supplied documentation, for example.

13.6 Reliability

a. Is the tool reliable in its operation, i.e. does it work consistently without frequent failures.

Requirements Management Tools Evaluation User Needs and Criteria

Andrew P. Gabb, Neelan Maheswaran, Alan M. Allwright

(DSTO-GD-0139)

DISTRIBUTION LIST

	Number of Copies
AUSTRALIA	
DEFENCE ORGANISATION	
S&T Program	
Chief Defence Scientist)	
FAS Science Policy)	1 shared copy
AS Science Corporate Management)	
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Control sheet
Counsellor, Defence Science, Washington	Doc Control sheet
Scientific Adviser to MRDC Thailand	Doc Control sheet
Director General Scientific Advisers and Trials)	1 shared copy
Scientific Adviser - Policy and Command)	
Navy Scientific Adviser	1 copy of Doc Control sheet and 1 distribution list
Scientific Adviser - Army	Doc Control sheet and 1 distribution list
Air Force Scientific Adviser	1
Director Trials	1
Director Aeronautical & Maritime Research Laboratory	1
Electronics and Surveillance Research Laboratory	
Director Electronics and Surveillance Research Laboratory	1
Chief Information Technology Division	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Computing Systems	1
Research Leader Command, Control and Communications	1
Executive Officer, Information Technology Division	Doc Control sheet
Head, Information Architectures Group	1
Head, C3I Systems Engineering Group	1
Head, Information Warfare Studies Group	1
Head, Software Engineering Group	1
Head, Trusted Computer Systems Group	Doc Control sheet
Head, Advanced Computer Capabilities Group	Doc Control sheet
Head, Systems Simulation and Assessment Group	Doc Control sheet
Head, Intelligence Systems Group	Doc Control sheet
Head Command Support Systems Group	1
Head CCIS Interoperability Lab	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control sheet
Head Information Management and Fusion Group	Doc Control sheet
Head, Human Systems Integration Group	Doc Control sheet
Author (Mr Andrew P. Gabb, C3ISE Group, ITD)	15
Publications and Publicity Officer, ITD	1

DSTO Library and Archives

Library Fishermens Bend	1
Library Maribyrnong	1
Library DSTOS	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control sheet

Forces Executive

Director General Maritime Development	Doc Control sheet
Director General Land Development	Doc Control sheet

Navy

SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000.	Doc Control sheet
Assistant Chief of Naval Staff - Materiel	1

Army

ABCA Office, G-1-34, Russell Offices, Canberra	4
Assistant Chief of the General Staff - Materiel	1

Air Force

Assistant Chief of the Air Force - Materiel	1

Intelligence Program

Deputy Director, Defence Intelligence Organisation	1
Library, Defence Signals Directorate	Doc Control sheet

Acquisition Program

Director General Communication Systems	4
Director, Information Management and Communications Engineering	1
First Assistant Secretary, Capital Equipment Programs	1
Director General Acquisition Planning	1
Director, Command, Communications and Computing Systems	1
Director, Project Management Education and Training	1
Head Industry & Procurement Infrastructure	1

Corporate Support Program (libraries)

OIC, Defence Regional Library, Canberra	1
Officer in Charge, Document Exchange Centre (DEC),	1
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1

Universities and Colleges

Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Deakin University, Serials Section (M list)), Deakin University Library,	1
Senior Librarian, Hargrave Library, Monash University	1
Librarian, Flinders University	1

Other Organisations

NASA (Canberra)	1
AGPS	1

State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations	
INSPEC: Acquisitions Section Institution of Electrical Engineers	1
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts	1
Documents Librarian, The Center for Research Libraries, US	1
Information Exchange Agreement Partners	
Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1
SPARES	10
Total number of copies:	**91**

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)
N/A

2. TITLE

Requirements Management Tools Evaluation - User Needs and Evaluation Criteria

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
Document (U)
Title (U)
Abstract (U)

4. AUTHOR(S)

Andrew P. Gabb
Neelan Maheswaran
Alan M. Allwright

5. CORPORATE AUTHOR
Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury SA 5108

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-GD-0139	AR-010-201	General Document	August 1997

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
N8315/8/4	DST94/383	N/A	44	Nil

13. DOWNGRADING/DELIMITING INSTRUCTIONS
N/A

14. RELEASE AUTHORITY
Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT
Approved for Public Release

16. DELIBERATE ANNOUNCEMENT
No Limitation

17. CASUAL ANNOUNCEMENT
No Limitation

18. DEFTEST DESCRIPTORS

Software Tools
Software Evaluation
Australian Defence Force

19. ABSTRACT
This paper discusses the users' needs and evaluation criteria for requirements management (RM) tools for use in front end systems engineering processes in the Australian Defence Organisation. Part 1 of the paper describes the user needs for RM tools; Part 2 describes the evaluation criteria, based on those needs, to be used for comparative evaluation of the tools. The purpose of the paper was to establish the basis for a systematic evaluation of possible tools.